

AVR112 : ブート領域なしデバイス用TWIブートローダ

要点

- 以下を持つ全てのATMEL AVRマイクロ コントローラ
 - ・ 2Kバイト フラッシュ メモリ
 - ・ ブート領域なし
 - ・ 2線TWI/USIインターフェース
- AVROSP適合
- 支援される全ての目的対象デバイス用の応用コード例
- プログラミングとデバッグにIAR workbenchを使用
- TWI主装置用コードドライバ
- TWI/USI従装置用コードドライバ
- 応用領域プログラミング
 - ・ どのプログラミング ツールもなし
 - ・ 応用領域完全消去
 - ・ 応用領域プログラミング
 - ・ 応用ソフトウェア版可読性
 - ・ ブートローダ ソフトウェア可読性

1. 序説

この応用記述はチップ上にブート領域を持たず、最低2Kバイトのフラッシュメモリを持つATMEL®の8ビット AVR®マイクロ コントローラに関してTWI通信チャネルとしてTWIまたはUSIハードウェアを用いるブートローダの実装を記述します。ホスト装置はUART経由でAVR開放ソース書き込み器(AVROSP:AVR Open Source Programmer)が走行するPCと通信するATmega2560に基づきます。ホスト装置はTWI主装置として、一方目的対象はTWI従装置として働きます。

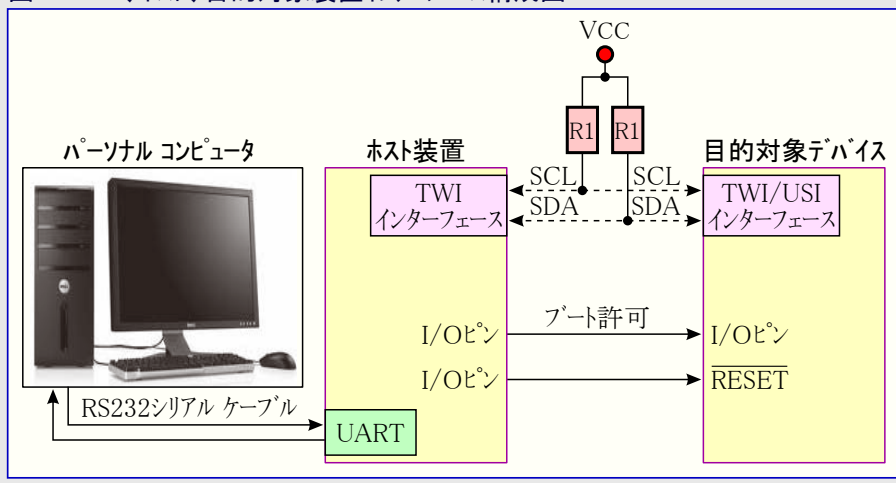
これは外部書き込み器を持たずに目的対象MCUへ応用コードを書き込み、ブートローダの版と応用の版の詳細を読むことを使用者に許します。

ダウンロードしたAVR112.zipファイルは試験応用の構築、従装置にそれを書き込み、試験応用の検証に必要な全ての例と有用物を含みます。

自己プログラミングについての一般的な情報に関しては「AVR109:自己プログラミング」応用記述を参照してください。

ハードウェア インターフェース構成図が図1-1.で図解されます。

図1-1. PC、ホスト、目的対象装置インターフェース構成図

8ビット ATMEL
マイクロ コントローラ

応用記述

本書は一般の方々の便宜のため有志により作成されたもので、ATMEL社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

2. 理屈

本章はマイクロコントローラのブートローダの詳細な記述を与えます。TWI/USI部署のより多くの詳細な情報についてはデータシートを参照してください。

2.1. マイクロコントローラのブートローダは何、そしてどう動くの？

一般的にマイクロコントローラをプログラミングするにはその目的対象デバイスを支援する書き込み器が必要です。高価なことを別にして、それは長いプログラミング時間、マイクロコントローラと書き込み器自身間の多すぎる相互接続のような、他に不利な点を持つかもしれません。プログラミングポートの非有効性のため、出荷された製品に対するファームウェア更新は飽き飽きする作業になります。コネクタで利用可能な通信インターフェースがある時に、ブートローダは便利になります。例え研究所であっても、各作業台に書き込み器を得ることが現実的でないかもしれず、最良の解決策はプログラムメモリ内にブートローダを既に一度書き込まれた試供マイクロコントローラを提供することです。

一旦ブートローダが書かれると、各々のインターフェース(UARTまたはTWI)は伝統的な書き込み器の使用なしで残りのフラッシュメモリを再書き換えするのに使用することができます。この応用記述では、ブートローダがTWIインターフェースと更新処理を検出するのに他のピンを使用します。最初にブートローダのファームウェアはBOOT PINの状態を検査し、Lowが見つかった場合、ブート領域に形態設定されたプログラムが実行を開始し、さもなければ応用領域に形態設定されたプログラムが実行を開始します。

図2-1は応用とブートの領域に分割されるフラッシュ領域での簡単な概念を与えます。

図2-1で指定されたアドレスがそれらのフラッシュに依存して各種目的対象デバイスで変わることにご注意してください。

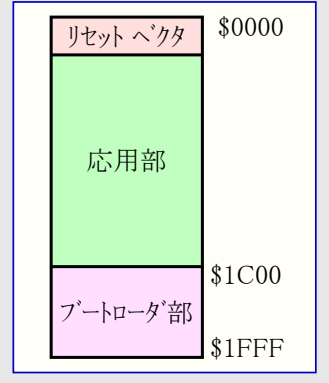
要約すると、ブートローダに関する段階は次の通りです。

段階1：リセット後、ブートローダはブートピンの状態を調べ、状態符号の流れに応じて、ブートローダ内で動くか、または応用へ飛びます。ブート領域内の時にブートローダのコードは到着データに対してTWIインターフェース指令の聴取を始めます。

段階2：受信した指令がフラッシュ更新の場合、受信したデータは書き込み器で書かれるように、プログラムメモリ(応用領域)に書かれます。

段階3：やって来るデータが一旦完全に書かれて、再び受信した指令が応用コード実行の場合、ブートローダは正規のプログラムの最初の命令へ飛びます。

図2-1. フラッシュメモリ割り当て



2.2. ブートローダの必要条件

TWIブートローダはハードウェア書き込み器でのどんな物理的な介入も不要にします。目的対象デバイスはホストとのいくつかのデータ接続の形態を持ちます。この応用記述ではI²C互換TWI通信が使用されます。デバイスがバスで他の装置との通信にTWIポートを使用する場合、追加の接続は全く不要です。ブートローダはブートローダファームウェアに予約されたプログラムメモリの上位側が必要です(図2-1をご覧ください)。プログラムメモリの残りは応用によって使用することができます。SRAMは同時ではなく、違う流れで使用されるため、予約される必要は全くありません。

2.3. 基本動作

ブートローダの基本的な機能はホストソフトウェアによって両方が既知の指令の組を受信して解釈し、そして実行することです。指令の組は応用記述で示されるよりもっと複雑に、またはより簡単にすることができますが、一般的にフラッシュメモリの読み込み、消去、書き込みの指令を支援しなければなりません。また、ブートローダコードから応用コードへ飛ぶ指令も支援します。この指令は代表的にプログラミング周回の最後で発行されます。

3. ハードウェア構成

本章はTWI主装置とTWI従装置の役割、それらの形態設定、その物理的な接続を記述します。

3.1. TWI主装置 - 説明

TWI主装置はUARTを通してAVROSP PCソフトウェアからの指令を受け入れます。受信した指令はTWI通信チャネルを経由してTWI従装置に渡って送られます。この応用記述では、TWI主装置としてSTK600開発基板上のATmega2560デバイスが使用され、詳細な接続は表3-1で記載されます。

3.2. TWI従装置 - 説明

TWI従装置はブートローダ経由でプログラミングされるべき応用領域を持つ目的対象デバイスです。TWI従装置はTWI通信チャネル経由での指令を受け入れ、予め定義された目的対象デバイス上での活動を実行します。この応用記述では、STK600開発基板上の目的対象デバイスとしてATtiny88デバイスが使用されます。

3.3. TWI主装置とTWI従装置の形態設定

本項はTWI主装置とTWI従装置が目的対象電圧、クロック周波数、ヒューズ設定などでどう形態設定されるかを記述します。

3.3.1. TWI主装置形態設定

デバイス : ATmega2560
 目的対象電圧 : 5V
 目的対象周波数 : 3.69MHz外部クリスタル
 ヒューズ設定 : \$FF(拡張)、\$19(上位)、\$FD(下位)

3.3.2. TWI従装置形態設定

従装置は以下の一覧からの何れかの装置になり得ます。従装置は周辺機能部署の有効性に基づく2つの一覧に分類されます。いくつかは組み込みTWIを持ち、その他はTWIインターフェースとして使用することができるUSIハードウェアを持ちます。「**TWIインターフェースを持つデバイス**」と「**USIインターフェースを持つデバイス**」項はその各々に一致するデバイスを簡単に記します。USIハードウェアはTWI従装置として形態設定され、より多くの詳細については「**AVR312: I²C従装置としてのUSI部の使い方**」応用記述を参照してください。例として、ATtiny88の形態設定が「**TWI従装置としてのATtiny88**」で記述されます。下の一覧内の他のデバイスについては「**ハードウェア構成**」頁とダウンロードしたzipフォルダ内の**Configuration Details.xls**エクセルシート資料の特定デバイス列を参照してください。

3.3.2.1. TWIインターフェースを持つデバイス

ATtiny48, ATtiny48A, ATtiny48PA, ATtiny88

3.3.2.2. USIインターフェースを持つデバイス

ATtiny2313A, ATtiny4313, ATtiny43U, ATtiny24A, ATtiny44A, ATtiny45/45V, ATtiny85/85V, ATtiny261A, ATtiny461A, ATtiny861A, ATtiny87, ATtiny167

3.3.2.3. TWI従装置としてのATtiny88

デバイス : ATtiny88
 目的対象電圧 : 5V
 目的対象周波数 : 8MHz内部RC発振器
 ヒューズ設定 : \$FE(拡張)、\$DD(上位)、\$EE(下位)

3.4. TWI主装置とTWI従装置間のハードウェア接続

試験構成はTWI主装置を持つ1つ目とTWI従装置を持つ2つ目で、2つのSTK600を含みます。支援される全ての目的対象デバイスについては「**ハードウェア構成**」頁とダウンロードしたzipフォルダ内の**Configuration Details.xls**エクセルシート資料内で記述されます。例としてTWI主装置(ATmega2560)とTWI従装置(ATtiny88)に対する接続詳細が下の表3-1.で言及されます。

表3-1. TWI主装置とTWI従装置間のハードウェア接続

TWI主装置としての ATmega2560	TWI従装置としての ATtiny88	説明
M_PD1_SDA	S_PC4_SDA	主装置SDA線(M_PD1_SDA)は従装置SDA線(S_PC4_SDA)に接続されます(TWI SDA線は外部的にプルアップされなければなりません)。
M_PD0_SCL	S_PC5_SCL	主装置SCL線(M_PD0_SCL)は従装置SCL線(S_PC5_SCL)に接続されます(TWI SCL線は外部的にプルアップされなければなりません)。
M_PC0_Enable	S_PB0	このピンはブートローダ許可(Low活性)を手助けし、ブートローダ開始時にブートローダによってポーリングされます。主装置のM_PC0_Enableピンが従装置のS_PB0ピンに接続されます。
M_PD1_RESET	S_PC6/RESET	主装置がこの線を用いて従装置をリセットします。 注: 従装置のデバッグWIREを禁止してください。主装置のM_PC1_RESETピンはS_PC6/RESETピンへ接続されます。
VCC	VCC	主装置と従装置の両方がVCCと接続されなければなりません。
GND	GND	主装置と従装置の両方に対する共通接地。
	LED0_PB1	LED0は従装置のPB1ピンに接続されます。
	LED1_PB3	LED1は従装置のPB3ピンに接続されます。
PCシリアルへのRS232 SPARE (直列変換器へのUSB使用禁止)		PCからのAVROSP指令を受け入れるため、主装置のシリアルポートをPCへ接続してください。

4. ソフトウェア構成

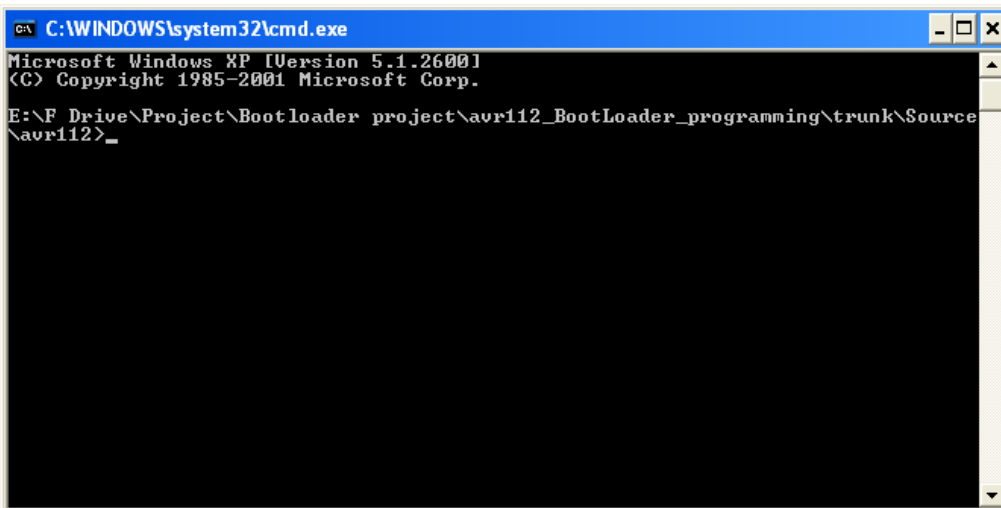
本章はPCソフトウェア、TWI主装置ファームウェア、TWI従装置ファームウェア、それと実演応用ソフトウェアの使用方法の詳細な記述を与えます。従装置応用領域の更新を実行するため、外部書き込み器を用いて“TWI_Master.a90”で書かれた主装置と“TWI_Slave.a90”で書かれた従装置を持たなければなりません。

“TWI_Slave.a90”プログラムは“TWI_Master”からの指令に応答し、“TWI_Master”はUARTを通してAVROSP Windows[®]ソフトウェア経由でPCからの指令を得ます。

4.1. PC側ソフトウェア

ここで、PCソフトウェアとしてAVROSPを使用し、ATmega2560はTWI主装置として形態設定されます。PCソフトウェアのAVROSPは(従装置に書かれるべき)HEXファイルをインポートし、それをより小さなパケットに分割し、そしてRS232インターフェース経由でデータを主装置へ送る、主要な機能を持ちます。

図4-1. AVROSPソフトウェア応用ウインドウ



AVROSPのより多くの詳細については「AVR911:AVR開放ソース書き込み器」応用記述を参照してください。

4.1.1. ブートローダ指令

この設計で使用されるブートローダ ファームウェアは目的対象デバイスのフラッシュ メモリへの読み込み、消去、書き込みに使用される指令の組を支援します。指令は表4-1.で示されます。

表4-1. フォトローダ指令

指令名	説明
TWIBL COMSETUP	COMポートをAVROSP通信用に形態設定。 以下の命令指示が表示されます。 Enter the COM_PORT_NUMBER 1 Status for device COM1: ----- Baud: 115200 Parity: None Data Bits: 8 Stop Bits: 1 Timeout: ON XON/XOFF: OFF CTS handshaking: OFF DSR handshaking: OFF DSR sensitivity: OFF DTR circuit: ON RTS circuit: ON
TWIBL HOSTSIG	TWIBL_PROG(ソフトウェア)とTWI_MASTER間の通信を保证するために、TWI_Masterの識票を読みます。
TWIBL UPDATE <ファイル名.hex>	フォトローダ経由でTWI従装置に応用(<ファイル名.hex>)を書き込みます。
TWIBL DIAG	TWI関係を除いて最後の指令に対する診断メッセージを表示します。
TWIBL EXEC	フォトローダから応用領域へ飛んで応用を走らせます。
TWIBL ERASE	リセット領域を除いて応用領域全体を消去します。
TWIBL BVERSION	フォトローダの版番号を読みます。
TWIBL AVERSION	応用の版番号を読みます。

4.2. TWI主装置ファームウェア

PC側AVROSPソフトウェアからの指令と従装置の応用HEXファイルを受け入れるように主装置ファームウェアが書かれます。望んだパケット受信後、それはTWI経由で従装置へ送られ、TWI従装置でのフラッシュ更新の実行を手助けします。

このファームウェアには各々UARTとTWIを用いてPCと目的対象デバイスの両方と通信する責任があります。

4.3. TWI従装置ファームウェア

TWI従装置ファームウェアはポート領域がなく、最低2Kバイトのフラッシュメモリを持ち、TWIとしてTWIかUSIのどちらかのハードウェアを持つ全てのAVRマイクロコントローラ用に開発されています。このフォトローダにはTWI主装置(ATmega2560)からのフラッシュ内容を受け取って、受信した指令に依存して必要な活動を実行する責任があります。

5. 始める前に

- ダウンロードしたzipフォルダ内のConfiguration Details.xlsエクセルシート資料のハードウェア構成の“MASTER PROGRAMMER”列で記載されたハードウェア接続準備をしてください。

5.1. TWI主装置のプログラミング

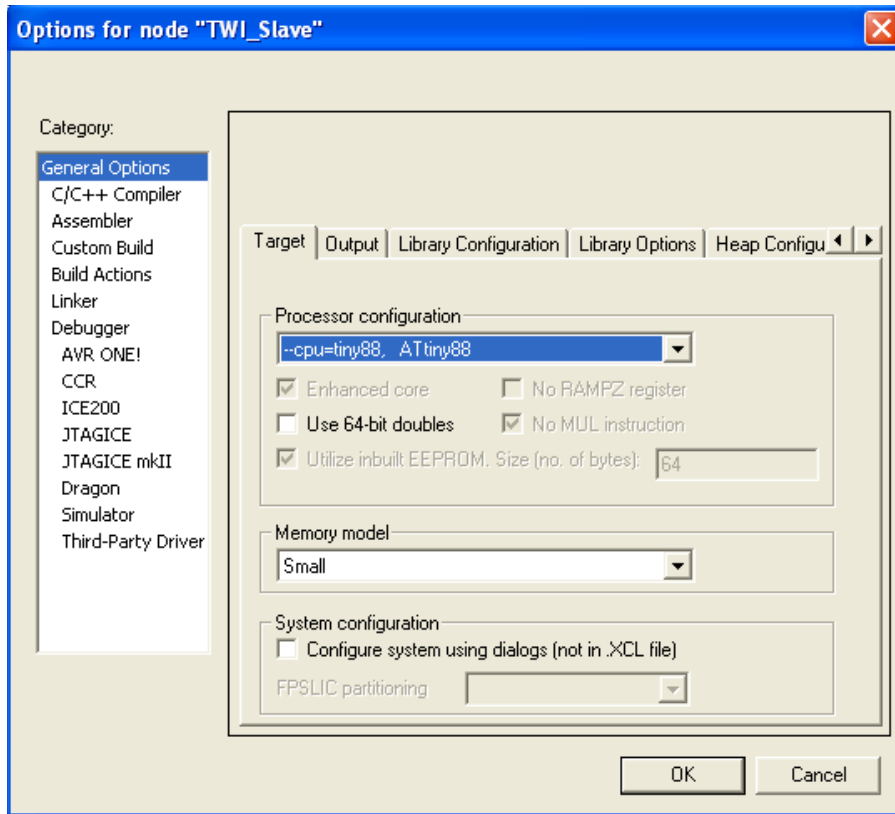
- ¥avr112_BootLoader_programming¥trunk¥Source¥avr112¥TWI_Master¥TWI_Master.ewwを開いてください。
- 全てを保存(Save All)
- 解除(Clean)
- 全てを再構築(Rebuild All)
- TWI_Master¥Release¥Exe¥TWI_Master.a90でATmega2560に書き込んでください。

5.2. TWI従装置のプログラミング

- ¥avr112_BootLoader_programming¥trunk¥Source¥avr112¥TWI_Master¥TWI_Slave.ewwを開いてください。
- 以下の2つの図で記載されるようにIAR Embedded Workbench® IDEでTWI従装置を選択してください。例えばここではTWI従装置としてATtiny88デバイスが選択されます。

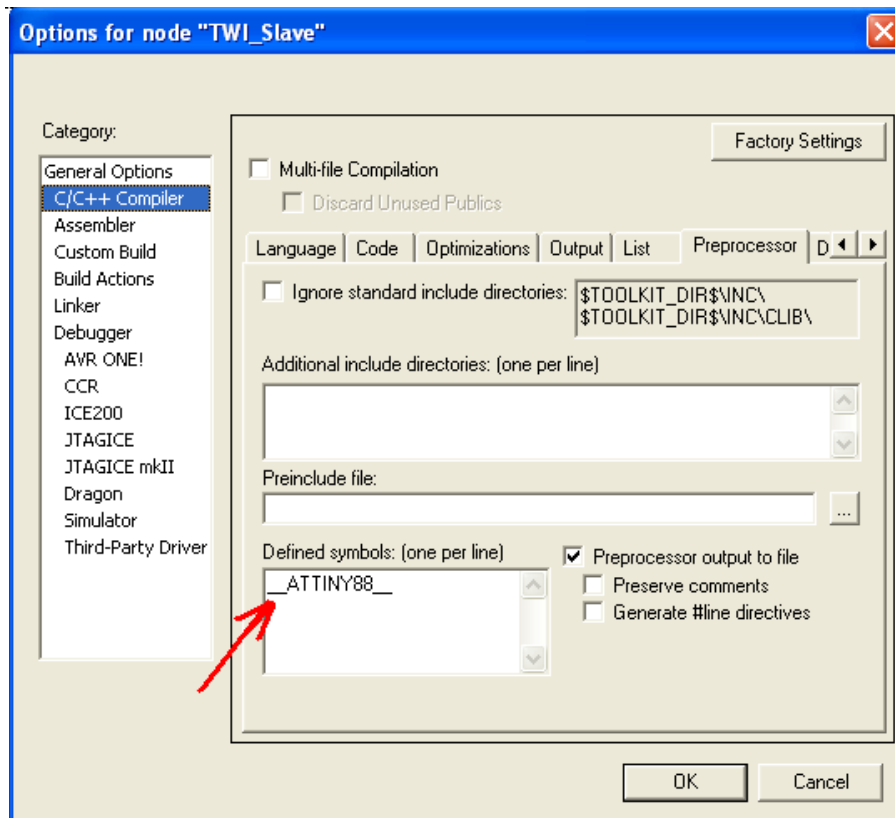
Go to Project⇒Options⇒Category⇒General Options⇒Targetタブ

図5-1. IAR Embedded Workbench IDE構成設定1



Project⇒Options⇒Category⇒C/C++ Compiler⇒Preprocessorタブ

図5-2. IAR Embedded Workbench IDE構成設定2



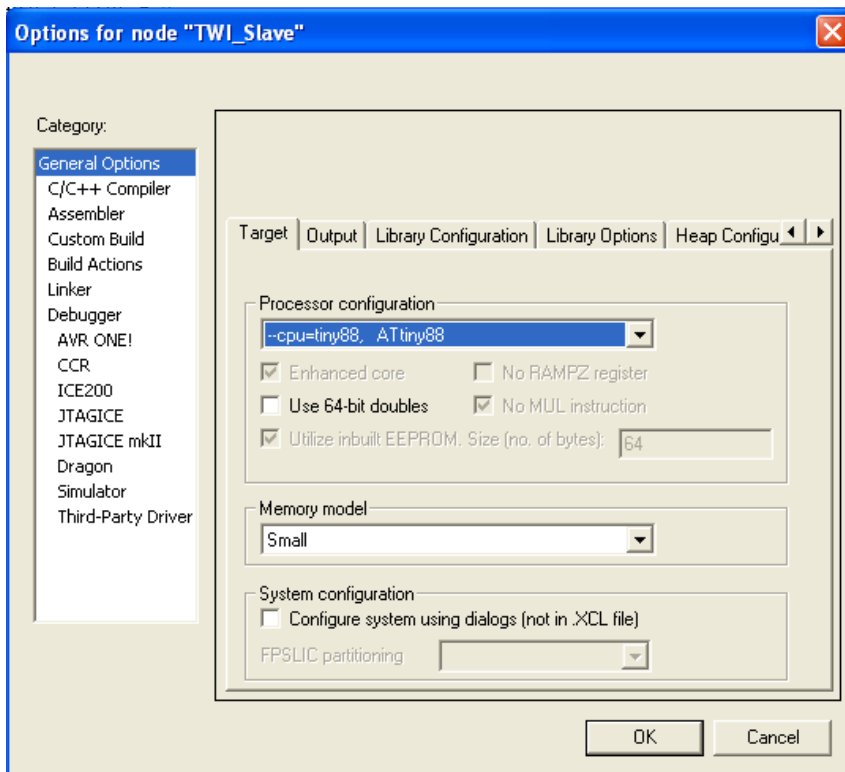
3. 全てを保存(Save All)
4. 解除(Clean)
5. 全てを再構築(Rebuild All)
6. 選択した従装置デバイスにTWI_Slave¥Release¥Exe¥TWI_Slave.a90を書き込んでください。

5.3. ブートローダ経由でプログラミングされるべきLED点滅応用のHEXファイル生成

1. `¥avr112_BootLoader_programming¥trunk¥Source¥avr112¥Demo_App1¥Demo_App1.eww`を開いてください。
2. 以下の2つの図で記載されるようにIAR Embedded Workbench® IDEでTWI従装置を選択してください。例えばここではTWI従装置としてATtiny88デバイスが選択されます。

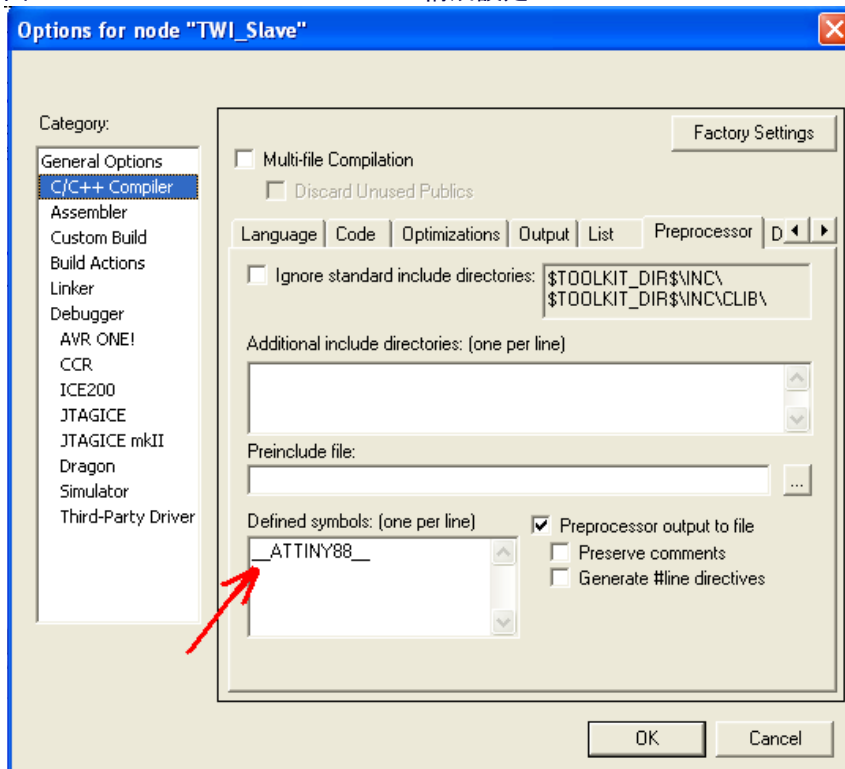
Go to Project⇒Options⇒Category⇒General Options⇒Targetタブ

図5-3. IAR Embedded Workbench IDE構成設定1



Project⇒Options⇒Category⇒C/C++ Compiler⇒Preprocessorタブ

図5-4. IAR Embedded Workbench IDE構成設定2



3. 全てを保存(Save All)
4. 解除(Clean)
5. 全てを再構築(Rebuild All)
6. 今やDemoApp1.hexの準備が整いました。

5.4. 機能試験

1. TWI主装置とTWI従装置の両方を通電してください(ホストブート準備可を示すためにLED0とLED1の両方が点灯するでしょう)。
2. "Start Cmd"ファイルをダブルクリックしてください。
3. TWIBLを入力すると、初回にCOM PORT NUMBERを問うので、主装置に接続されたCOMポート番号を入力してください。これはCOMPORT設定を形態設定します。これはTWIBL指令も一覧にします。
4. DemoApp1.hexで応用領域をプログラミングする(書き込む)ために、TWIBL UPDATE DemoApp1.hexを入力してください。
5. プログラミングされたコードを走らせるために、TWIBL EXECを入力してください。目的対象従装置がDemoApp1.hexで成功裏にプログラミングされているなら、これはLED0を点滅するでしょう。
6. プログラミングされた応用を消去するため、TWIBL ERASEを入力してください。
7. EEPROMのアドレス\$05から応用の版番号を得るため、TWIBL AVERSIONを入力してください。応用領域がプログラミングされて(書かれて)いない場合に何が起きますか？
8. ブートローダの版番号を得るため、TWIBL BVERSIONを入力してください。

同様に、上で記載した手続きを用いて、LED1を点滅する別の応用のHEXファイルを生成してTWI従装置デバイスを再書き込みしてください。



Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
USA
TEL (+1)(408) 441-0311
FAX (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited

Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
TEL (+852) 2245-6100
FAX (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
TEL (+49) 89-31970-0
FAX (+49) 89-3194621

Atmel Japan

141-0032 東京都品川区
大崎1-6-4
新大崎勸業ビル 16F
アトメル ジャパン合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

© 2011 Atmel Corporation. 全権利予約済

ATMEL®、ATMELロゴとそれらの組み合わせ、それとAVR®、AVR®ロゴとその他はATMEL Corporationの登録商標または商標またはその付属物です。Windows®とその他は米国とその他の国に於いてMicrosoft Corporationの登録商標です。他の用語と製品名は一般的に他の商標です。

お断り: 本資料内の情報はATMEL製品と関連して提供されています。本資料またはATMEL製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。ATMELのウェブサイトに位置する販売の条件とATMELの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、ATMELはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえATMELがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してATMELに責任がないでしょう。ATMELは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。ATMELはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、ATMEL製品は車載応用に対して適当ではなく、使用されるべきではありません。ATMEL製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© HERO 2013.

本応用記述はATMELのAVR112応用記述(doc8079.pdf Rev.8079A-08/11)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。